Tytuł projektu: DomFinder - Platforma ogloszeń nieruchomości

Krótki opis dzialania projektu

DomFinder to interaktywna platforma ogloszeń nieruchomości, która umożliwia użytkownikom dodawanie własnych ofert sprzedaży czy wynajmu nieruchomości oraz przeglądanie ogloszeń innych. Dzięki prostemu i intuicyjnemu interfejsowi, użytkownicy mogą łatwo filtrować wyniki wyszukiwania i zarządzać swoimi ogłoszeniami.

Autorzy projektu:

- Julian Arciszewski,
- Magdalena Zerebilo

Specyfikacja wykorzystanych technologii

Backend:

Technologia - C# .NET 8.0

Baza danych - MS SQL Server

Lista używanych pakietów NuGet wraz z ich wersjami:

- Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore (8.0.3)

- Microsoft.AspNetCore.Identity.EntityFrameworkCore (8.0.3)

- Microsoft.AspNetCore.Identity.UI (8.0.3)

- Microsoft.AspNetCore.SignalR.Client (8.0.5)

- Microsoft.AspNetCore.SignalR.Common (8.0.5)

- Microsoft.AspNetCore.SignalR.Protocols.Json (8.0.5)

- Microsoft.EntityFrameworkCore.SqlServer (8.0.3)

- Microsoft.EntityFrameworkCore.Tools (8.0.4)

- Microsoft.VisualStudio.Web.CodeGeneration.Design (8.0.2)

Instrukcja pierwszego uruchomienia projektu

1. Należy zainstalować wymagane narzędzia i biblioteki za pomoca komendy:

dotnet restore

2. Następnie skonfigurować baze danych, wykonując migracje:

Update-Database

3. Uruchom aplikacje:

dotnet run

4. Możemy zarejestrować się po uruchomieniu aplikacji lub skorzystać z automatycznie stworzonych kont:

Admin:
login: admin@example.com
hasło: Admin@123

Zwykły użytkownik:
login: test@mail.com
hasło: Test@123

Opis struktury projektu

Projekt DomFinder składa się z kilku głównych komponentów:

- **Controllers -** kontrolery zarządzają logiką aplikacji,
- **Models** - modele reprezentuja strukture danych aplikacji,
- **Views -** widoki odpowiadają za prezentację danych użytkownikowi,
- **Wwwroot -** katalog zawierający statyczne zasoby, takie jak pliki CSS, JS i obrazy,
- **Data -** pliki zwiazane z baza danych i seedowaniem danych,
- **Program.cs -** główny plik konfiguracyjny aplikacji.

1. **Modele**

Modele aplikacji znajduja sie w katalogu Models. Poniżej znajduje się lista modeli wraz z ich krótkim opisem oraz polami:

- **Property** - *model reprezentujący nieruchomość.*

PropertyID - Identyfikator nieruchomości,

Title - Tytuł ogłoszenia,

Price - Cena nieruchomości,

Area - Powierzchnia nieruchomoci,

Rooms - Liczba pokoi,

Description - Opis nieruchomości,

City – Miasto,

District – Dzielnica,

Street – Ulica,

BuildingType - Rodzaj zabudowy,

Floor – Piętro,

TotalFloors - Liczba pięter,

YearBuilt - Rok budowy,

OwnershipType - Forma własności,

AvailableFrom - Data dostępności,

Rent - Czynsz,

AdvertiserType - Typ ogłoszeniodawcy,

EnergyCertificate - Certyfikat energetyczny.

Internet - Internet,

CableTV - Telewizja kablowa,

Phone - Telefon,

AntiBurglaryBlinds - Czy sa rolety antywłamaniowe,

Monitoring - Czy jest monitoring,

AlarmSystem - Czy jest system alarmowy,

Intercom - Czy jest domofon,

ClosedArea - Czy teren jest zamkniety.

Furnished - Czy umeblowane,

WashingMachine - Czy jest pralka,

Fridge - Czy jest lodówka,

Stove - Czy jest kuchenka,

Television - Czy jest telewizor,

Dishwasher - Czy jest zmywarka,

Oven - Czy jest piekarnik,

UserId - Identyfikator użytkownika,

UserName - Nazwa uzytkownika,

UserPhoneNumber - Numer telefonu użytkownika,

ImagePaths - Ścieżki do zdjęc nieruchomoci.

- **UserProfileViewModel -** *model widoku profilu użytkownika*.

User - obiekt użytkownika (IdentityUser).

Properties - lista nieruchomości użytkownika.

- **PropertyViewModel -** *model widoku nieruchomości*.

Property - obiekt nieruchomości.

IsEditable - czy nieruchomość jest edytowalna przez aktualnego użytkownika.

- **ErrorViewModel -** *model widoku błędu*.

RequestId - Identyfikator zadania.

ShowRequestId: - Czy pokazać identyfikator zadania.

- **ChatMessage -** model wiadomości czatu.

Id - Identyfikator wiadomości.

SenderId - Identyfikator nadawcy.

ReceiverId - Identyfikator odbiorcy.

Message – Treść wiadomości.

Timestamp - Znacznik czasu wiadomosci.

ChatRoomId - Identyfikator pokoju czatu.

- **ChatRoomViewModel -** model widoku pokoju czatu.

ChatRoomId - Identyfikator pokoju czatu.

UserId - Identyfikator użytkownika.

UserEmail - Email użytkownika.

- **ChatViewModel -** model widoku czatu.

CurrentUserId - Identyfikator aktualnego uzytkownika.

OtherUserId - Identyfikator drugiego uzytkownika.

OtherUserName - Nazwa drugiego uzytkownika.

Messages - Lista wiadomosci w czacie.

2. Kontrolery

Kontrolery aplikacji znajduja sie w katalogu Controllers. Poniżej znajduje się lista kontrolerów wraz z metodami:

- **HomeController**

Metody:

Index() - Wyświetla strone glowna aplikacji. [GET]

Privacy() - Wyświetla strone z polityka prywatnosci. [GET]

Error() - Wyświetla strone błędu. [GET]

- **PropertiesController**

Metody:

Index(string city, string district, decimal? minPrice, decimal? maxPrice, decimal? minArea, decimal? maxArea, int? minRooms, int? maxRooms, string transactionType) - wyświetla liste wszystkich nieruchomości. [GET]

Details(int? id) - Wyswietla szczegoly wybranej nieruchomosci. [GET]

Create() - Wyswietla formularz dodawania nowej nieruchomosci. [GET]

Create(Property property, List<IFormFile> Photos) - Dodaje nowa nieruchomosc do bazy danych. [POST]

Edit(int? id) - Wyswietla formularz edycji nieruchomosci. [GET]

Edit(int id, Property property, List<IFormFile> Photos) - Aktualizuje dane nieruchomości w bazie danych. [POST]

Delete(int? id) - Wyswietla strone usuwania nieruchomosci. [GET]

DeleteConfirmed(int id) - Usuwa wybrana nieruchomosc z bazy danych. [POST]

- **UsersController**

UserProfile(string id) - Wyswietla profil uzytkownika. [GET]

- **AdminController**

Users() - Wyswietla liste wszystkich uzytkownikow. [GET]

ManageUser(string id) - Wyswietla szczegoly uzytkownika i umozliwia zarzadzanie nim. [GET]

ToggleAdminRole(string id) - Dodaje lub odbiera role administratora. [POST]

BanUser(string id, int days) - Zablokowuje uzytkownika na okreslona liczbe dni. [POST]

UnbanUser(string id): Odbanowuje uzytkownika. [POST]

- **ChatController**

Chat(string userId): Wyswietla czat z uzytkownikiem. [GET]

Messages(): Wyswietla wszystkie wiadomosci czatu. [GET]

SendMessage(string receiverId, string message): Wysyla wiadomosc czatu. [POST]

- **System użytkowników**

Role - System obsluguje role uzytkownikow, takie jak admin i user.

Funkcje:

- Uzytkownicy zalogowani moga dodawac, edytowac i usuwac swoje ogloszenia.

- Administratorzy moga zarzadzac wszystkimi uzytkownikami i ich ogloszeniami.

- Goscie moga tylko przegladac ogloszenia, bez mozliwosci dodawania, edytowania lub usuwania.

- Role uzytkownikow sa nadawane i odbierane przez administratorow za pomoca metody ToggleAdminRole w AdminController.

- **Najciekawsze funkcjonalności**

Filtrowanie ogłoszeń - Użytkownicy mogą filtrować ogłoszenia wedlug róznych kryteriow, takich jak cena, lokalizacja, czy data dodania.

System wiadomosci -Umożliwia komunikacje miedzy uzytkownikami w celu ustalenia szczegolow dotyczacych ogloszen.

3. Widoki

Poniżej znajduje się lista widoków aplikacji:

- **Admin**
  - DeleteUser.cshtml
  - ManageUser.cshtml
  - Users.cshtml

- **Chat**
  - Chat.cshtml
  - Messages.cshtml

- **Home**
  - Index.cshtml
  - Privacy.cshtml

- **Properties**
  - Create.cshtml
  - Delete.cshtml
  - Details.cshtml
  - Edit.cshtml
  - Index.cshtml

- **Shared**
  - _Layout.cshtml
  - _LoginPartial.cshtml
  - _ValidationScriptsPartial.cshtml
  - Error.cshtml

- **Users**
  - UserProfile.cshtml